What Is Claimed Is:

1   1.   A method for deleting a node from a skip list, wherein the skip list

2   is lock-free, which means that the skip list can be simultaneously accessed by

3   multiple processes without requiring the processes to perform locking operations,

4   the method comprising:

5   receiving a reference to a target node to be deleted from the skip list;

6   marking a next pointer in the target node to indicate that the target node is

7   deleted, wherein the next pointer contains the address of an immediately

8   following node in the skip list, wherein the marking operation does not destroy the

9   address of the immediately following node, and wherein the marking operation is

10   performed atomically and thereby without interference from other processes; and

11   atomically modifying the next pointer of an immediately preceding node in

12   the skip list to point to the immediately following node in the skip list, instead of

13   pointing to the target node, thereby splicing the target node out of the skip list.

1   2.   The method of claim 1, wherein after the target node is spliced out

2   of the skip list, the method further comprises modifying the next pointer of the

3   target node so that the next pointer remains marked but points to the immediately

4   preceding node instead of the immediately following node in the skip list.

1   3.   The method of claim 1,

2   wherein a node in the skip list can possibly be a tall node that includes one

3   or more higher-level next pointers, wherein a given higher-level next pointer

4   contains the address of the immediately following node in the skip list that

5   reaches or exceeds the height of the given higher-level next pointer; and

16

6   wherein if the target node is a tall node, the method further comprises

7   similarly marking and splicing around higher-level next pointers in the target

8   node.

1   4.   The method of claim 1, wherein marking the next pointer to

2   indicate that the target node is deleted involves setting a "deleted bit" in the next

3   pointer.

1   5.   The method of claim 1, wherein marking the next pointer to

2   indicate that the target node is deleted involves:

3   creating a special node with a deleted type, which points to the

4   immediately following node in the skip list; and

5   atomically replacing the next pointer with a pointer to special node.

1   6.   The method of claim 1, further comprising inserting a new node

2   into the skip list by:

3   locating a node immediately preceding the new node in the skip list,

4   wherein locating the immediately preceding node involves maintaining a

5   predecessor array, wherein for each level of the skip list, the predecessor array

6   contains a pointer to the node immediately preceding the new node in the skip list;

7   locating a node immediately following the new node in the skip list,

8   wherein locating the immediately following node involves maintaining a

9   successor array, wherein for each level of the skip list, the successor array

10   contains a pointer to the immediately following node; and

11   splicing the new node into the skip list by,

12   setting the next pointer for the new node to point to the

13   immediately following node,

17

14	atomically updating the next pointer of the immediately
15	preceding node to point to the new node, and
16	if the new node is a tall node, similarly splicing higher-
17	level next pointers associated with the new node.

1	7. The method of claim 1, further comprising removing a highest
2	priority node from the skip list through a constant time operation, wherein the
3	head node of the skip list points to the highest priority node for ease of deletion,
4	and wherein keys for nodes are chosen to achieve this ordering.

1	8. The method of claim 1, further comprising periodically performing
2	a garbage-collection operation to reclaim deleted nodes that have become
3	unreachable.

1	9. The method of claim 1, wherein the target node includes:
2	a key that contains a priority value for the node in the skip list;
3	a value field that contains or points to data associated with the node;
4	a next pointer that contains the address of an immediately following node
5	in the skip list; and
6	zero or more higher-level next pointers, wherein a given higher-level next
7	pointer contains the address of the immediately following node in the skip list that
8	reaches or exceeds the height of the given next pointer.

1	10. A computer-readable storage medium storing instructions that
2	when executed by a computer cause the computer to perform a method for
3	deleting a node from a skip list, wherein the skip list is lock-free, which means

18

4   that the skip list can be simultaneously accessed by multiple processes without

5   requiring the processes to perform locking operations, the method comprising:

6   receiving a reference to a target node to be deleted from the skip list;

7   marking a next pointer in the target node to indicate that the target node is

8   deleted, wherein the next pointer contains the address of an immediately

9   following node in the skip list, wherein the marking operation does not destroy the

10   address of the immediately following node, and wherein the marking operation is

11   performed atomically and thereby without interference from other processes; and

12   atomically modifying the next pointer of an immediately preceding node in

13   the skip list to point to the immediately following node in the skip list, instead of

14   pointing to the target node, thereby splicing the target node out of the skip list.

1   11.   The computer-readable storage medium of claim 10, wherein after

2   the target node is spliced out of the skip list, the method further comprises

3   modifying the next pointer of the target node so that the next pointer remains

4   marked but points to the immediately preceding node instead of the immediately

5   following node in the skip list.

1   12.   The computer-readable storage medium of claim 10,

2   wherein a node in the skip list can possibly be a tall node that includes one

3   or more higher-level next pointers, wherein a given higher-level next pointer

4   contains the address of the immediately following node in the skip list that

5   reaches or exceeds the height of the given higher-level next pointer; and

6   wherein if the target node is a tall node, the method further comprises

7   similarly marking and splicing around higher-level next pointers in the target

8   node.

1   13.   The computer-readable storage medium of claim 10, wherein

2 marking the next pointer to indicate that the target node is deleted involves setting

3 a "deleted bit" in the next pointer.

1   14.   The computer-readable storage medium of claim 10, wherein

2 marking the next pointer to indicate that the target node is deleted involves:

3   creating a special node with a deleted type, which points to the

4 immediately following node in the skip list; and

5   atomically replacing the next pointer with a pointer to special node.

1   15.   The computer-readable storage medium of claim 10, wherein the

2 method further comprises inserting a new node into the skip list by:

3   locating a node immediately preceding the new node in the skip list,

4 wherein locating the immediately preceding node involves maintaining a

5 predecessor array, wherein for each level of the skip list, the predecessor array

6 contains a pointer to the node immediately preceding the new node in the skip list;

7   locating a node immediately following the new node in the skip list,

8 wherein locating the immediately following node involves maintaining a

9 successor array, wherein for each level of the skip list, the successor array

10 contains a pointer to the immediately following node; and

11   splicing the new node into the skip list by,

12   setting the next pointer for the new node to point to the

13   immediately following node,

14   atomically updating the next pointer of the immediately

15   preceding node to point to the new node, and

16   if the new node is a tall node, similarly splicing higher-

17   level next pointers associated with the new node.

20

1   16.   The computer-readable storage medium of claim 10, wherein the

2   method further comprises removing a highest priority node from the skip list

3   through a constant time operation, wherein the head node of the skip list points to

4   the highest priority node for ease of deletion, and wherein keys for nodes are

5   chosen to achieve this ordering.

1   17.   The computer-readable storage medium of claim 0, wherein the

2   method further comprises periodically performing a garbage-collection operation

3   to reclaim deleted nodes that have become unreachable.

1   18.   The computer-readable storage medium of claim 10, wherein the

2   target node includes:

3   a key that contains a priority value for the node in the skip list;

4   a value field that contains or points to data associated with the node;

5   a next pointer that contains the address of an immediately following node

6   in the skip list; and

7   zero or more higher-level next pointers, wherein a given higher-level next

8   pointer contains the address of the immediately following node in the skip list that

9   reaches or exceeds the height of the given next pointer.

1   19.   An apparatus for deleting a node from a skip list, wherein the skip

2   list is lock-free, which means that the skip list can be simultaneously accessed by

3   multiple processes without requiring the processes to perform locking operations,

4   the apparatus comprising:

5   a receiving mechanism configured to receive a reference to a target node to

6   be deleted from the skip list;

21

7   a marking mechanism configured to mark a next pointer in the target node

8 to indicate that the target node is deleted, wherein the next pointer contains the

9 address of an immediately following node in the skip list, wherein the marking

10 operation does not destroy the address of the immediately following node, and

11 wherein the marking operation is performed atomically and thereby without

12 interference from other processes; and

13   a splicing mechanism configured to atomically modify the next pointer of

14 an immediately preceding node in the skip list to point to the immediately

15 following node in the skip list, instead of pointing to the target node, thereby

16 splicing the target node out of the skip list.

1   20. The apparatus of claim 19, wherein after the target node is spliced

2 out of the skip list, the splicing mechanism is configured to modify the next

3 pointer of the target node so that the next pointer remains marked but points to the

4 immediately preceding node instead of the immediately following node in the skip

5 list.

1   21. The apparatus of claim 19,

2   wherein a node in the skip list can possibly be a tall node that includes one

3 or more higher-level next pointers, wherein a given higher-level next pointer

4 contains the address of the immediately following node in the skip list that

5 reaches or exceeds the height of the given higher-level next pointer; and

6   wherein if the target node is a tall node, the marking mechanism and the

7 splicing mechanism are similarly configured to mark and splice around higher-

8 level next pointers in the target node.

22

1   22.   The apparatus of claim 19, wherein while marking the next pointer

2   to indicate that the target node is deleted, the marking mechanism is configured to

3   set a "deleted bit" in the next pointer.

1   23.   The apparatus of claim 19, wherein while marking the next pointer

2   to indicate that the target node is deleted, the marking mechanism is configured

3   to:

4   create a special node with a deleted type, which points to the immediately

5   following node in the skip list; and to

6   atomically replace the next pointer with a pointer to special node.

1   24.   The apparatus of claim 19, further comprising an insertion

2   mechanism configured to insert a new node into the skip list, wherein the

3   insertion mechanism is configured to:

4   locate a node immediately preceding the new node in the skip list, wherein

5   locating the immediately preceding node involves maintaining a predecessor

6   array, wherein for each level of the skip list, the predecessor array contains a

7   pointer to the node immediately preceding the new node in the skip list;

8   locate a node immediately following the new node in the skip list, wherein

9   locating the immediately following node involves maintaining a successor array,

10   wherein for each level of the skip list, the successor array contains a pointer to the

11   immediately following node; and to

12   splice the new node into the skip list by,

13   setting the next pointer for the new node to point to the

14   immediately following node,

15   atomically updating the next pointer of the immediately

16   preceding node to point to the new node, and

23

17   if the new node is a tall node, similarly splicing higher-

18   level next pointers associated with the new node.

1   25.   The apparatus of claim 19, further comprising a popping

2   mechanism configured to remove a highest priority node from the skip list

3   through a constant time operation, wherein the head node of the skip list points to

4   the highest priority node for ease of deletion, and wherein keys for nodes are

5   chosen to achieve this ordering.

1   26.   The apparatus of claim 19, further comprising a garbage collection

2   mechanism configured to periodically perform a garbage-collection operation to

3   reclaim deleted nodes that have become unreachable.

1   27.   The apparatus of claim 19, wherein the target node includes:

2   a key that contains a priority value for the node in the skip list;

3   a value field that contains or points to data associated with the node;

4   a next pointer that contains the address of an immediately following node

5   in the skip list; and

6   zero or more higher-level next pointers, wherein a given higher-level next

7   pointer contains the address of the immediately following node in the skip list that

8   reaches or exceeds the height of the given next pointer.

24